(a) selecting a page having "free" status and changing the status of said page to "cacheable"; and

(b) dynamically allocating a block of said page to be accessed by said processor;

if a further block of memory is required for storage of data local to a specific processor then:

(a) if a page having "cacheable" status has an unallocated block, dynamically allocating said block of said page to be accessed by said processor; and

(b) if no block of a page having "cacheable" status is available then selecting a page having "free" status, changing the status of said page to "cacheable" and dynamically allocating a block of said page to be accessed by said processor;

al

if a block of memory is required for storage of data to be accessed by more than one processor then:

(a) selecting a page having "free" status and changing the status of said page to "non-cacheable"; and

(b) dynamically allocating a block of said page to be accessed by any processor;

if a further block of memory is required for storage of data to be accessed by more than one processor then:

(a) if a page having "non-cacheable" status has an unallocated block, dynamically allocating said block of

17

A

said page to be accessed by any processor; and

(b) if no block of a page having "non-cacheable" status is available then selecting a page having "free" status, changing the status of said page to "non-cacheable" and dynamically allocating a block of said page to be accessed by any processor;

retaining a page record as to the status of each page; and

retaining an allocation record as to which blocks of a page have been allocated.

9 ~~10~~. (New) A method according to claim 8 ~~9~~ wherein if an allocated block is no longer required, the allocation record is amended to discard the allocation of the block.

a1

10 ~~11~~. (New) A method according to claim 8 ~~9~~ wherein if no blocks on a page of memory having "cacheable" or "non-cacheable" status are allocated, the status of said page is changed to "free".

11 ~~12~~. (New) A method according to claim 9 ~~10~~ wherein if no blocks on a page of memory having "cacheable" or "non-cacheable" status are allocated, the status of said page is changed to "free".

12 ~~13~~. (New) A method according to claim 8 ~~9~~ wherein the step of discarding the allocation of a block allocated from a page having "cacheable" status comprises the step of discarding the data of the block.

1 ~~14~~. (New) A microprocessor system comprising:

18

A

at least two processors, each processor having a cache memory; and

a system memory which is divided into pages, each of which initially has a "free" status and is subdivided into unallocated blocks;

wherein the system is responsive to a first request for allocation of memory space of cacheable or non-cacheable type, by:

dynamically allocating a block of memory from a page of "free" status, the system thereafter changing the status of said page from "free" to "cacheable" or "non-cacheable" as the case may be; and

is responsive to a further request for allocation of memory space of cacheable or non-cacheable type, by:

dynamically allocating a block of memory from a page of appropriate status; or

if such a block is unavailable, dynamically allocating a block from a page having "free" status, the system thereafter changing the status of said page from "free" to "cacheable" or "non-cacheable" as the case may be.

a1

2 ~~15~~. (New) A system according to claim 1 ~~14~~ wherein the system is responsive to a request that an allocated block of memory is to be discarded.

3 3 ~~16~~. (New) A system according to claim 1 ~~14~~, the system further being responsive to a request to discard a block in that if said block is the only allocated block on the relevant page of memory then the system changes the status of said page to "free".

4 ~~17~~. (New) A system according to claim 1 ~~14~~ wherein the cache memory of each processor is divided into lines.

19

A

5 ~~18~~. (New) A system according to claim 1 ~~14~~ wherein the size of the blocks of the system memory is a whole multiple of the size of the lines.

6 ~~19~~. (New) A system according to claim 2 ~~15~~ wherein the cache memory of each processor is divided into lines.

a1

7 ~~20~~. (New) a system according to claim 6 ~~19~~ wherein ~~Wherein~~ the size of the blocks of the system memory is a whole multiple of the size of the lines.

YEE 5/22/00

REMARKS

A Request For One-Month Extension plus the fee is enclosed.

In the office action, the Examiner has objected that the priority claim in the declaration is incorrect. We enclose with this response a new declaration with the correct priority date inserted.

Claims 1-8 have been canceled. New claims 9 and 14 correspond substantially to original claims 1 and 3 respectively.

The Examiner objected to original claim 3 as failing to comply with 35 U.S.C. § 112. It is submitted that this objection is now moot in view of the cancellation of original claim 3 and the insertion of new claim 14.

The Examiner also objected to original claims 1 and 8 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 4,885,680 (Anthony) in view of U.S. Patent No. 5,897,660 (Reinders), U.S. Patent No. 5,321,834 (Weiser) and U.S. Patent No. 5,075,848 (Lai).

New claim 14 explicitly recites that the system comprises a system memory which is "divided into pages, each of which initially has a "free" status and is subdivided unto unallocated blocks."

A